

METHOD OF AND APPARATUS FOR MULTIMEDIA PROCESSING, AND COMPUTER PRODUCT

BACKGROUND OF THE INVENTION

5 1) Field of the Invention

The present invention relates to a technology for increasing the efficiency of managing multimedia information based on names.

2) Description of the Related Art

10 Manufacturers use various kinds of names such as development code names, drawing numbers, part names, firmware names, and software names in a series of operation processes including product planning, designing, manufacturing, and field support. In each operation process, the operation is performed based on names as keys. Further, in each operation process, a considerable amount of text
15 information, image information, and acoustic information is used as multimedia information. See, for example, Japanese Patent Application Laid-Open No. 2002-63033 for details.

However, since it is difficult to perform checking of duplication inside and outside facilities and setting of management rule thoroughly when giving a name or
20 using multimedia information, checking is apt to be omitted and it is difficult to manage the multimedia information.

SUMMARY OF THE INVENTION

It is an object of the present invention to solve at least the problems in the
25 conventional technology.

The multimedia processing apparatus according to one aspect of the present invention includes a setting unit that sets a specific element from an installation space where each element to be given a name is hierarchically expressed, a generating unit that generates a name space ontology based on name information, wherein the name space ontology is a group of name candidates with the set element as a top level, and a linking unit that links each name constituting the name space ontology with multimedia information.

The multimedia processing method according to another aspect of the present invention includes setting a specific element from an installation space where each element to be given a name is hierarchically expressed, generating a name space ontology based on name information, wherein the name space ontology is a group of name candidates with the set element as a top level, and linking each name constituting the name space ontology with multimedia information.

The computer program according to still another aspect of the present invention realizes the multimedia processing method according to the present invention on a computer.

The other objects, features and advantages of the present invention are specifically set forth in or will become apparent from the following detailed descriptions of the invention when read in conjunction with the accompanying drawings.

20

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram representing the structure of a system according to an embodiment of the present invention;

Fig. 2 is a schematic of the installation space in the system according to the embodiment;

25

Fig. 3 is a schematic of the name information in the system according to the embodiment;

Fig. 4 is a schematic of the name space (ontology) in the system according to the embodiment;

5 Fig. 5 illustrates the operation of the system according to the embodiment;

Fig. 6 illustrates the operation of the system according to the embodiment;

Fig. 7 illustrates part table list information according to the embodiment;

Fig. 8 is a flowchart of the operation of the system according to the embodiment;

10 Fig. 9 is a flowchart of the registration processing shown in Fig. 8;

Fig. 10 is a flowchart of the name obtaining processing shown in Fig. 9;

Fig. 11 is a flowchart of the neural network ontology generation processing shown in Fig. 9;

15 Fig. 12 is a flowchart of the neural network ontology generation processing shown in Fig. 9;

Fig. 13 is a flowchart of the layer depth determination processing shown in Fig. 12;

Fig. 14 is a flowchart of the categorizing processing shown in Fig. 12;

Fig. 15 is a flowchart of the search/reference processing shown in Fig. 8;

20 Fig. 16 is a flowchart of the image query processing shown in Fig. 12;

Fig. 17 illustrates the outline of the operation based on the neural network according to the embodiment;

Fig. 18 is a schematic of the neural network according to the embodiment;

Fig. 19 is a schematic of the neural network according to the embodiment;

25 Fig. 20 is a schematic of the security gateway for each operation process

according to the embodiment;

Fig. 21 illustrates the linking operation according to the embodiment;

Fig. 22 is a schematic of the fuzzy logic according to the embodiment; and

Fig. 23 is a block diagram representing the structure of a system according to
5 a modified embodiment.

DETAILED DESCRIPTION

Exemplary embodiments of a computer program, a multimedia processing
apparatus, and a multimedia processing method according to the present invention are
10 explained in detail below with reference to the accompanying drawings.

Fig. 1 is a block diagram representing the structure of a system according to
an embodiment of the present invention. A multimedia processing apparatus 100
executes a naming service of providing names using ontology, and a management
service of managing multimedia information corresponding to the names.

15 A communicating unit 101 of the multimedia processing apparatus 100
controls communications between the multimedia processing apparatus 100, a
registration system client 300, and a search/reference system client 400 via a network
200 based on a predetermined communication protocol.

The registration system client 300, which is a computer terminal that registers
20 name information and ontology information, accesses the multimedia processing
apparatus 100 via the network 200.

The search/reference system client 400, which is a computer terminal to
search for names and multimedia information, accesses the multimedia processing
apparatus 100 via the network 200.

25 An installation space and a name space (ontology) are the key points. The

installation space represents elements that are installed on a specific apparatus, in a hierarchical structure (including a class, and a first layer to the lowest layer) as shown in Fig. 2. In the example shown in Fig. 2, a server is the class.

The first layer includes a locker and a power supply. The second layer
5 includes a PCB01 (printed circuit board 01) that belongs to the locker of the first layer, a PCB00 (printed circuit board 00) that belongs to the power supply of the first layer, and the like.

The third layer includes a microprocessor that belongs to the PCB01 of the second layer, a flash memory that belongs to the PCB00 of the second layer, and the
10 like.

The fourth layer includes a cooling fan that belongs to the microprocessor of the third layer, and the like. The fifth layer includes a heat sink that belongs to the cooling fan of the fourth layer, and the like. The lowest layer includes a pipe that belongs to the heat sink of the fifth layer, and the like. In this manner, the elements
15 that constitute the apparatus are expressed hierarchically in the installation space.

The elements that constitute an apparatus are usually given their own names. Fig. 3 is a schematic of name information according to the embodiment. For the apparatus, a DCN (development code name), a part name, and an OTN (other name) belong to the same domain. The DCN is provided in relation to the development of a
20 corresponding apparatus, and includes an FMN (family name), a KSN (machine type name), a SUN (apparatus name), and a UNN (unit name).

The part name includes a CPN (component name), a DVN (device name), and an ELN (element name). The ELN is the minimum unit of the name.

Names are expressed in a name space (ontology) shown in Fig. 4 based on
25 a concept system that represents vertical and subordination relationship between

concepts. The top level is an item to be conceptualized (i.e., an item to be given a name). The first layer to the lowest layer have elements (i.e. names) based on the above relationship, respectively.

In the example shown in Fig. 5 the PCB00 is positioned at the top level
5 among the part table list of the server, and the microprocessor and the flash memory are at the lower level positions, respectively.

A registration control unit 102 in Fig. 1 controls registration of information at each database. An ontology editing unit 103 has a function of editing the ontology. A name obtaining unit 104 has a function of obtaining a name candidate. An ontology
10 generating unit 105 has a function of automatically generating the ontology (see Fig. 4) based on a method of a neural network, a fuzzy logic, or a genetic algorithm.

A search/reference processing unit 106 has a function of carrying out search/reference in response to the search/reference request from the search/reference system client 400. A multimedia information database 110 stores
15 multimedia information (image information, video information, acoustic information, and various kinds of documents (for example, information concerning standards, design, manufacturing, purchasing, maintenance, etc.)).

A word dictionary information database 111 stores word dictionary information as candidates for names. An ontology information database 112 stores ontology
20 information that is generated by the ontology generating unit 105. A name information database 113 stores name information that is obtained by the name obtaining unit 104.

A name space information database 114 stores name space information. An installation space information database 115 stores part table list information (see
25 Fig. 7) corresponding to the installation space shown in Fig. 2. A schema vocabulary

information database 116 stores schema vocabulary information.

The operation of the system according to the embodiment is explained below with reference to flowcharts shown in Fig. 8 to Fig. 16.

At step SA1 shown in Fig. 8, the registration control unit 102 decides whether
5 there is a registration request from the registration system client 300. If a result of the decision is "No," the search/reference processing unit 106 decides whether there is a search/reference request from the search/reference system client 400 at step SA2. If the result of the decision is "No," the steps SA1 and SA2 are repeated until the result of the decision becomes "Yes."

10 When there is a registration request from the registration system client 300, the registration control unit 102 sets "Yes" as a result of the decision at the step SA1. At step SA3, the registration processing is executed.

Specifically, at step SB1 of the flowchart shown in Fig. 9, the registration control unit 102 decides whether there is a new registration request. In this example,
15 the registration control unit 102 sets "Yes" as a result of the decision. At step SB2, the name obtaining unit 104 executes the name obtaining processing.

Specifically, at step SC1 of the flowchart shown in Fig. 10, the name obtaining unit 104 decides whether reading of the name information is assigned from the web (not shown in the figure) that uses the registration system client 300.

20 When a result of the decision at the step SC1 is "No," the name obtaining unit 104 reads the word dictionary information (i.e., name information) from the word dictionary information database 111 at step SC3. On the other hand, when the result of the decision at the step SC1 is "Yes," the name obtaining unit 104 reads the assigned contents (i.e., name information) from the web at step SC2.

25 The name information that the name obtaining unit 104 reads at the step SC3

or SC2 is the development code name, the part name, and the other name within the domain (i.e., area) shown in Fig. 3. The development code name includes the FMN, the KSN, the SUN, and the UNN. The part name includes the CPN, the DVN, and the ELN.

5 The name information is ruled as an XML (extensible markup language) name space in the W3C (worldwide web consortium).

 At step SC4, the name obtaining unit 104 decides whether an extension is assigned to the name information read at the step SC3 or SC2. The extension provides the name with extendibility, and this includes hyphen, ### (a reserved
10 character), prefix, and suffix, etc.

 When a result of the decision at the step SC4 is "Yes," the name obtaining unit 104 decides at step SC5 whether the user who assigned the extension coincides with the owner of the name information having the extension.

 When a result of the decision at the step SC5 is "No," the name obtaining unit
15 104 makes a display unit (not shown in the figure) of the registration system client 300 display an error message at step SC7. On the other hand, when the result of the decision at the step SC5 is "Yes," the name obtaining unit 104 approves the assignment of the extension at step SC6, and executes the processing at step SC8.

 When the result of the decision at the step SC4 is "No," the name obtaining
20 unit 104 at the step SC8 collates the name information already registered in the name information database 113 with the name information read at the step SC3 SC2. At step SC9, the name obtaining unit 104 decides whether both name information coincide with each other as a result of the collation at the step SC8.

 When a result of the decision at the step SC9 is "No," the name obtaining unit
25 104 decides at step SC10 whether the name information exists in other domains.

When a result of the decision at the step SC10 is "Yes," the name obtaining unit 104 at step SC11 decides whether to use the name information by adding a domain name to the header of the DCN or the ELM (element name) shown in Fig. 3 and connecting the name information with PAT/. When a result of the decision at the step SC11 is "No,"
5 the name obtaining unit 104 makes a decision at step SC15.

On the other hand, when the result of the decision at the step SC10 is "No," or when a result of the decision at the step SC11 is "Yes," the name obtaining unit 104 at step SC12 decides whether the lower constituent element (such as a part name) of the name information is unique. In this example, the name obtaining unit 104 sets "Yes"
10 as a result of the decision at the step SC12. At step SC13, the name obtaining unit 104 registers the name information into the name information database 113.

On the other hand, when the result of the decision at the step SC12 is "No," the name obtaining unit 104 at step SC14 makes the display unit of the registration system client 300 display an error message (for example, "Please change the name,
15 as there exists the same constituent element.").

When the result of the decision at the step SC9 is "Yes," or when the result of the decision at the step SC11 is "No," the name obtaining unit 104 decides at the step SC15 whether to search for other similar word. When a result of the decision at the step SC15 is "No," the name obtaining unit 104 at step SC16 makes the display unit of
20 the registration system client 300 display a re-input message (for example, "The name already exists. Please input other name.").

On the other hand, when the result of the decision at the step SC15 is "Yes," the name obtaining unit 104 at step SC17 searches the name information database 113, and decides whether a similar word that is analogous with the input name
25 information exists. When a result of the decision at the step SC17 is "No," the name

obtaining unit 104 makes a decision at the step SC1. When the result of the decision at the step SC17 is "Yes," the name obtaining unit 104 at step SC18 urges the input of the similar word from the web.

Referring back to the step SB3 in Fig. 9, the registration control unit 102
5 decides whether to generate the ontology (see Fig. 4) based on an approach 1. In this example, the registration control unit 102 sets "No" as a result of the decision at the step SB3. The approach 1 is a method of using an existing drawing number or an existing article number. On the other hand, when the result of the decision at the step SB3 is "Yes," the registration control unit 102 at step SB8 generates the ontology
10 based on the existing drawing number or the existing article number, and stores the ontology into the ontology information database 112.

At step SB4, the registration control unit 102 decides whether to generate the ontology (see Fig. 4) based on an approach 2. In this example, the registration control unit 102 sets "No" as a result of the decision at the step SB4. The approach 2
15 is a method of using a vendor specification provided from an external vendor or the like. When the result of the decision at the step SB4 is "Yes," the registration control unit 102 at step SB9 generates the ontology based on the vendor specification provided from the external vendor or the like, and stores the ontology into the ontology information database 112.

20 At step SB5, the registration control unit 102 decides whether to generate the ontology (see Fig. 4) based on an approach 3. In this example, the registration control unit 102 sets "No" as a result of the decision at the step SB5. The approach 3 is a method of using the ontology editing unit 103. When the result of the decision at the step SB5 is "Yes," the registration control unit 102 at step SB10 generates the
25 newly prepared ontology into the ontology information database 112.

At step SB6, the registration control unit 102 decides whether to generate the ontology based on an approach 4. In this example, the registration control unit 102 sets "Yes" as a result of the decision at the step SB6. The approach 4 is a method of using a neural network.

5 At step SB11, the ontology generating unit 105 executes the generation of the neural network ontology. As shown in Fig. 17, it is possible to apply the neural network ontology generation to any name of hardware, firmware, and software.

In the example shown in Fig. 17, the ontology generating unit 105 determines a development code name (for example, "Eagle") of the server, and determines a
10 development code name of a lower unit of Eagle. It is possible to avoid duplication of the determined name with any other existing names on a brochure both inside and outside the company.

Specifically, at step SD1 in a flowchart shown in Fig. 11, the ontology
generating unit 105 reads the part table list information shown in Fig. 7 from the
15 installation space information database 115.

At step SD2, the ontology generating unit 105 determines an "apparatus
name" as a name to be conceptualized, based on a user's instruction from the
registration system client 300. At step SD3, the ontology generating unit 105 decides
whether the desired name exists based on the user's instruction from the registration
20 system client 300.

When a result of the decision at the step SD3 is "Yes," the ontology
generating unit 105 at step SD4 makes the user input the desired name from the web
using the registration system client 300. When the result of the decision at the step
SD3 is "No," the ontology generating unit 105 at step SD5 decides whether the item to
25 be conceptualized is a ubiquitous web. In this example, the ontology generating unit

105 sets "No" as a result of the decision. On the other hand, when the result of the decision at the step SD5 is "Yes," the ontology generating unit 105 at step SD13 sets the ubiquitous web to the top level of the name space (see Fig. 4).

At step SD6, the ontology generating unit 105 decides whether the item to be
5 conceptualized is a family name or a machine type name. In this example, the ontology generating unit 105 sets "No" as a result of the decision. On the other hand, when the result of the decision at the step SD6 is "Yes," the ontology generating unit 105 at step SD14 sets the family name or the machine type name to the top level of the name space.

10 At step SD7, the ontology generating unit 105 decides whether the item to be conceptualized is an apparatus name. When a result of the decision at the step SD7 is "No," the ontology generating unit 105 at step SD8 decides whether the item to be conceptualized is a unit name. In this example, the ontology generating unit 105 sets "No" as a result of the decision. On the other hand, when the result of the decision at
15 the step SD8 is "Yes," the ontology generating unit 105 at step SD16 sets the unit name to the top level of the name space.

At step SD9, the ontology generating unit 105 decides whether the item to be conceptualized is a component name. In this example, the ontology generating unit 105 sets "No" as a result of the decision. On the other hand, when the result of the
20 decision at the step SD9 is "Yes," the ontology generating unit 105 at step SD17 sets the component name to the top level of the name space.

At step SD10, the ontology generating unit 105 decides whether the item to be conceptualized is a device name. In this example, the ontology generating unit 105 sets "No" as a result of the decision. On the other hand, when the result of the
25 decision at the step SD10 is "Yes," the ontology generating unit 105 at step SD18 sets

the device name to the top level of the name space.

At step SD11, the ontology generating unit 105 decides whether the item to be conceptualized is an element name. When a result of the decision at the step SD11 is "Yes," the ontology generating unit 105 at step SD19 sets the element name to the top level of the name space.

On the other hand, when the result of the decision at the step SD11 is "No," the ontology generating unit 105 at step SD12 sets other name to the top level of the name space.

In this example, the ontology generating unit 105 sets "Yes" as a result of the decision at the step SD7. At step SD15, the ontology generating unit 105 sets the apparatus name to the top level of the name space. In the flowchart shown in Fig. 12, at step SD20, the ontology generating unit 105 presents one candidate of name information to the user via the registration system client 300.

At step SD21, the ontology generating unit 105 decides whether the user likes the name information set as a candidate of the apparatus name, based on the user's operation.

When a result of the decision at the step SD21 is "No," the ontology generating unit 105 at step SD22 makes the user input the desired name from the web using the registration system client 300.

On the other hand, when the result of the decision at the step SD21 is "Yes," the ontology generating unit 105 at step SD23 allocates a group of the item to be conceptualized (the apparatus, in this case) to the name information that the user likes or to the input name information. At step SD24, the ontology generating unit 105 reads the name information of the item to be conceptualized, from the name information database 113.

At step SD25, the ontology generating unit 105 decides whether the name information read at the step SD24 is in the same group. When a result of the decision at the step SD25 is "No," the ontology generating unit 105 makes a decision at step SD17. On the other hand, when the result of the decision at the step SD25 is "Yes,"
5 the ontology generating unit 105 at step SD26 gives the same group ID to the name information.

At step SD27, the ontology generating unit 105 decides whether the presentation of the name information candidate at the step SD20 has all ended. In this example, the ontology generating unit 105 sets "No" as a result of the decision at
10 the step SD27, and executes the processing at the step SD20. Thereafter, the ontology generating unit 105 repeats the processing at the step SD20 to SD27, thereby to determine a candidate of the name information in the group.

When the result of the decision at the step SD27 is "Yes," the ontology generating unit 105 at step SD28 executes the layer depth determination processing to
15 determine the layer depth of the name space (ontology) generated based on the candidates of the name information, in the neural network (see Fig. 18).

Specifically, at step SE1 in a flowchart shown in Fig. 13, the ontology generating unit 105 takes out one piece of name information as an element of the group. At step SE2, the ontology generating unit 105 decides whether there is a firing
20 in the first layer of the name space (ontology) shown in Fig. 4. When a result of the decision at the step SE2 is "Yes," the ontology generating unit 105 at step SE3 sets a weight to the first layer.

When the result of the decision at the step SE2 is "No," the ontology generating unit 105 at step SE4 decides whether there is a firing in the second layer
25 shown in Fig. 4. When a result of the decision at the step SE4 is "Yes," the ontology

generating unit 105 at step SE5 sets a weight to the second layer.

Thereafter, the ontology generating unit 105 executes the processing at the steps SE2 and SE3 for the third layer afterward. At step SE6, the ontology generating unit 105 sets a weight to the lowest layer. At step SE7, the ontology generating unit 105 decides whether the taking out of the elements (i.e., name information) from the group is completed. In this example, the ontology generating unit 105 sets "No" as a result of the decision at the step SE7, and executes the processing at the step SE1 afterward.

When the result of the decision at the step SE7 is "Yes," the ontology generating unit 105 at step SE8 ends the learning based on the neural network. At step SE9, the ontology generating unit 105 determines the depth of the largest layer among the layers in which there is a firing, and stores the name space information of the hierarchical structure shown in Fig. 4 into the name space information database 114.

Referring back to step SD29 in Fig. 12, the ontology generating unit 105 executes the categorization processing to categorize the name information (concept). Specifically, at step SF1 in the flowchart shown in Fig. 14, the ontology generating unit 105 sets a table of "is-a" and "Part-of" that show mutual relationship of the concept. The "is-a" means that the concept expressed by two pieces of name information in the name space (see Fig. 3, Fig. 5, and Fig. 6) is in the lateral relationship. The "Part-of" means that the concept expressed by two pieces of name information is in the vertical relationship.

At step SF2, the ontology generating unit 105 reads the name space information (see Fig. 4) from the name space information database 114. At step SF3, the ontology generating unit 105 takes out pieces of name information that belong to

the same domain and the same layer, from the name space shown in Fig. 4.

At step SF4, the ontology generating unit 105 decides whether there is a firing in the "is-a" in the neural network. When a result of the decision at the step SF4 is "Yes," the ontology generating unit 105 at step SF5 categorizes the name information
5 in the "is-a."

On the other hand, when the result of the decision at the step SF4 is "No," the ontology generating unit 105 at step SF6 decides whether there is a firing in the "Part-of" in the neural network. When a result of the decision at the step SF6 is "Yes," the ontology generating unit 105 at step SF7 categorizes the name information in the
10 "Part-of." At step SF8, the ontology generating unit 105 reflects the firing state into the table.

At step SF9, the ontology generating unit 105 decides whether the processing of all the name information has ended in the name space information from the name space information database 114. In this example, the ontology generating unit 105
15 sets "No" as a result of the decision at the step SF9. Thereafter, the ontology generating unit 105 repeats the processing at the steps SF3 to SF9, thereby to form a meaning network (for example, a network as shown in Fig. 19) of the name information.

When the result of the decision at the step SF9 is "Yes," the ontology
20 generating unit 105 at step SF10 registers the categorized ontology shown in Fig. 4 into the ontology information database 112.

At step SB21 of the flowchart shown in Fig. 9, the registration control unit 102 reads the schema vocabulary information from the schema vocabulary information database 116. At step SB22, the registration control unit 102 describes the ontology
25 information stored in the ontology information database 112, in the ontology description

language (based on the resource description framework (RDF)), and registers this ontology information into the ontology information database 112.

At step SB23, the registration control unit 102 sets the operation process shown in Fig. 20. The operation process is a series of process that includes product
5 planning, designing, hardware (firmware)/software/manual development/printing and binding, production arrangement, purchasing, manufacturing, testing, inspection, marketing, sales promotion, taking orders, arranging, delivery, local procurement, and field support.

In each step of the operation process, names generated based on the above
10 method (a development code, for example) are used. Fig. 5 and Fig. 6 illustrate examples of the operation process being used.

At step SB24 of the flowchart shown in Fig. 9, the registration control unit 102 sets a security gate as shown in Fig. 20 to each step of the operation process. This security gate limits a range of names that can be searched for/referred to, depending
15 on the step of the operation process.

At step SB25, as shown in Fig. 21, the registration control unit 102 links each element (name) of the name space (ontology) shown in Fig. 4 with a corresponding piece of the multimedia information (i.e., image information, video information, acoustic information, and various kinds of documents (for example, information concerning
20 standards, design, manufacturing, purchasing, maintenance, etc.)). At step SB26, the registration control unit 102 stores the multimedia information linked with the names, into the multimedia information database 110.

When a result of the decision at the step SB6 shown in Fig. 9 is "No," the registration control unit 102 generates the ontology based on a genetic algorithm (GA).
25 When a result of the decision at step SB1 is "No," that is, when it is necessary to

correct the ontology, the registration control unit 102 at step SB12 decides whether to correct the ontology (see Fig. 4) generated based on the approach 1. In this example, the registration control unit 102 sets "No" as a result of the decision at the step SB12.

On the other hand, when the result of the decision at the step SB12 is "Yes,"
5 the registration control unit 102 at step SB17 corrects the ontology based on the existing drawing number or the existing article number.

At step SB13, the registration control unit 102 decides whether to correct the ontology (see Fig. 4) generated based on the approach 2. In this example, the registration control unit 102 sets "No" as a result of the decision at the step SB13. On
10 the other hand, when the result of the decision at the step SB13 is "Yes," the registration control unit 102 at step SB18 corrects the ontology based on the vendor specification provided from the external vendor or the like.

At step SB14, the registration control unit 102 decides whether to correct the ontology (see Fig. 4) generated based on the approach 3. In this example, the
15 registration control unit 102 sets "No" as a result of the decision at the step SB14. On the other hand, when the result of the decision at the step SB14 is "Yes," the registration control unit 102 at step SB19 corrects the ontology.

At step SB15, the registration control unit 102 decides whether to correct the ontology generated based on the approach 4. In this example, the registration control
20 unit 102 sets "No" as a result of the decision at the step SB15. On the other hand, when the result of the decision at the step SB15 is "Yes," the registration control unit 102 at step SB20 corrects the ontology based on the neural network. At step SB16, the registration control unit 102 corrects the ontology generated at step SB7, based on the genetic algorithm.

25 When the search/reference system client 400 makes a search/reference

request, the search/reference processing unit 106 sets "Yes" as a result of the decision at the step SA2 shown in Fig. 8. At step SA4, the search/reference processing unit 106 carries out the search/reference processing.

Specifically, at step SG1 in the flowchart shown in Fig. 15, the
5 search/reference processing unit 106 decides whether to use a text query as a search key. When a result of the decision at the step SG1 is "Yes," the search/reference processing unit 106 at step SG2 makes the search/reference system client 400 input the text query.

At step SG7, the search/reference processing unit 106 decides whether to
10 carry out the security search based on the security gate explained with reference to Fig. 20. In this example, the search/reference processing unit 106 sets "Yes" as a result of the decision at the step SG7. When the result of the decision at the step SG7 is "No," the search/reference processing unit 106 execute the processing at step SG9.

At step SG8, the search/reference processing unit 106 checks the security
15 gate (see Fig. 20) based on the operation process corresponding to the searcher of the search/reference system client 400.

At step SG9, the search/reference processing unit 106 searches the
multimedia information database 110 for multimedia information based on the search
key (the text query in this example), and retrieves the name linked with the multimedia
20 information from the ontology information database 112.

The search/reference processing unit 106 checks whether the result of the
search is limited by the security gate. When the result of the search is limited by the
security gate, the search/reference processing unit 106 decides that there is no result
of the search. At step SG10, the search/reference processing unit 106 makes the
25 display unit of the search/reference system client 400 display the result of the search

(i.e., multimedia information and name).

On the other hand, when the result of the decision at the step SG1 is "No," the search/reference processing unit 106 at step SG3 decides whether to use an image query as a search key. In this example, the search/reference processing unit 106 sets
5 "No" as a result of the decision at the step SG3.

At step SG5, the search/reference processing unit 106 decides whether to use an acoustic query as a search key. When a result of the decision at the step SG5 is "No," the search/reference processing unit 106 makes a decision at the step SG1

When the result of the decision at the step SG5 is "Yes," the search/reference
10 processing unit 106 at step SG6 makes the search/reference system client 400 input the acoustic query.

At the step SG7, the search/reference processing unit 106 decides whether to carry out the security search as described above. In this example, the search/reference processing unit 106 sets "Yes" as a result of the decision at the step
15 SG7.

At the step SG8, the search/reference processing unit 106 checks the security gate based on the operation process corresponding to the searcher of the search/reference system client 400.

At the step SG9, the search/reference processing unit 106 searches the
20 multimedia information database 110 for multimedia information based on the search key (the acoustic query in this example), and retrieves the name linked with the multimedia information from the ontology information database 112.

The search/reference processing unit 106 checks whether the result of the search is limited by the security gate. When the result of the search is limited by the
25 security gate, the search/reference processing unit 106 at the step SG10 makes the

display unit of the search/reference system client 400 display the result of the search (i.e., multimedia information and name).

On the other hand, when the result of the decision at the step SG3 is "Yes," the search/reference processing unit 106 at step SG4 executes the image query processing. Specifically, at step SH1 in the flowchart shown in Fig. 16, the
5 search/reference processing unit 106 decides whether to input a keyword as a search key. In this example, the search/reference processing unit 106 sets "Yes" as a result of the decision at the step SH1. At step SH2, the search/reference processing unit 106 makes the search/reference system client 400 input a keyword (for example, a
10 large yellow flower) corresponding to the image query.

On the other hand, when the result of the decision at the step SH1 is "No," the search/reference processing unit 106 at step SH3 makes the search/reference system client 400 display a plurality of image patterns as search key candidates, and makes the search/reference system client 400 select one candidate.

At step SH4, the search/reference processing unit 106 makes the
15 search/reference system client 400 display the image patterns corresponding to the step SH3 or SH2. At step SH5, the search/reference processing unit 106 makes the search/reference system client 400 select an image pattern. At step SH6, the search/reference processing unit 106 decides whether the image pattern selected at
20 the step SH5 is a target pattern.

When a result of the decision at the step SH6 is "No," the search/reference processing unit 106 executes the processing at the step SH4 afterward. When the result of the decision at the step SH6 is "Yes," at step SG7 of the flowchart shown in Fig. 15, the search/reference processing unit 106 decides whether to carry out the
25 security search. In this example, the search/reference processing unit 106 sets "Yes"

as a result of the decision at the step SG7.

At step SG8, the search/reference processing unit 106 checks the security gate based on the operation process corresponding to the searcher of the search/reference system client 400.

5 At step SG9, the search/reference processing unit 106 searches the multimedia information database 110 for multimedia information based on the search key (in this example, the image pattern that is set as the target pattern at the step SH6), and retrieves the name linked with the multimedia information from the ontology information database 112.

10 The search/reference processing unit 106 checks whether the result of the search is limited by the security gate. When the result of the search is limited by the security gate, the search/reference processing unit 106 decides that there is no result of the search. At step SG10, the search/reference processing unit 106 makes the display unit of the search/reference system client 400 display the result of the search
15 (i.e., multimedia information and name).

As explained above, according to the present embodiment, the system generates the ontology (see Fig. 4) as a group of name candidates, with an element set in the installation space (see Fig. 2) as a top level (see Fig. 4) based on the name information obtained in advance. The system links each name that constitutes this
20 ontology with the multimedia information. Therefore, it is possible to improve the efficiency of managing names and multimedia information.

According to the present embodiment, the system generates the ontology according to the element that is set to the top level of the name space (see Fig. 4). Therefore, it is possible to change the ontology to be generated according to the
25 element.

According to the present embodiment, the user selects whether the name based on the generated ontology is to be used as it is. Therefore, it is possible to reflect user's idea in naming.

According to the present embodiment, the system collates name information already obtained, thereby to check duplication of names, at the time of obtaining name information. Therefore, it is possible to avoid wasteful naming, and it is possible to further improve the management efficiency.

According to the present embodiment, the system checks duplication of names within the domain to which the name information belongs. Therefore, it is possible to use duplicated names between other domains.

According to the present embodiment, the system obtains name information having an extension equipped with extension function. Therefore, it is possible to increase extendibility of naming.

According to the present embodiment, the system sets the security gate that limits a range of names that can be searched for/referred to, according to the environment in which the name is used (i.e., the operation process), as shown in Fig. 20. Therefore, it is possible to increase the security level.

According to the present embodiment, the system executes the search for a name corresponding to the ontology, and multimedia information that is linked with the name, in the search/reference processing. The system outputs a result of the search corresponding to the security gate shown in Fig. 20. Therefore, there is an effect that it is possible to improve the efficiency of search for names and multimedia information.

A computer program for realizing the functions of the multimedia processing apparatus 100 may be recorded onto a computer-readable recording medium 600 shown in Fig. 23. A computer 500 shown in Fig. 23 reads the program recorded on

the recording medium 600, and executes the program to realize the functions.

The computer 500 includes a central processing unit (CPU) 510 that executes the program, an input unit 520 such as a keyboard and a mouse, a read-only memory (ROM) 530 that stores various kinds of data, a random-access memory (RAM) 540
5 that stores operation parameters and the like, a reading unit 550 that reads the program from the recording medium 600, an output unit 560 such as a display and a printer, and a bus 570 that connects between the units of the apparatus.

The CPU 510 reads the program recorded on the recording medium 600 via the reading unit 550, and executes the program, thereby to realize the functions. The
10 recording medium 600 includes an optical disk, a flexible disk, and a hard disk.

The neural network and the genetic algorithm are used to generate the ontology. It is also possible to use other methods (such as, a fuzzy logic shown in Fig. 22) to generate the ontology.

As explained above, according to one aspect of the present invention, the
15 system generates the ontology as a group of name candidates, with an element set in the installation space as a top level based on the name information obtained in advance. The system links each name that constitutes this ontology with the multimedia information. Therefore, it is possible to improve the efficiency of managing names and multimedia information.

20 According to another aspect of the present invention, the system sets the security gate that limits a range of names that can be searched for/referred to, according to the environment in which the name is used. Therefore, there is an effect that it is possible to increase the security level.

According to still another aspect of the present invention, the system executes
25 the search for a name corresponding to the name space ontology, and multimedia

information that is linked with the name, and outputs a result of the search corresponding to the security gate. Therefore, there is an effect that it is possible to improve the efficiency of search for names and multimedia information.

Although the invention has been described with respect to a specific
5 embodiment for a complete and clear disclosure, the appended claims are not to be thus limited but are to be construed as embodying all modifications and alternative constructions that may occur to one skilled in the art which fairly fall within the basic teaching herein set forth.